

# Units 1–4: Problem-solving methodology

Integral to the VCE Computing Study Design is a methodology for systematically creating solutions to problems, needs and opportunities. As applied in this study design the methodology comprises four stages: analysis, design, development and evaluation. For each of these stages there is a typical set of activities, as shown in Figure 1. Specific details of the scope of the problem-solving methodology are provided in the introduction to relevant areas of study. Note: when creating solutions, this methodology can be applied as a single stage-by-stage problem-solving process or to each iteration of an agile problem-solving process.

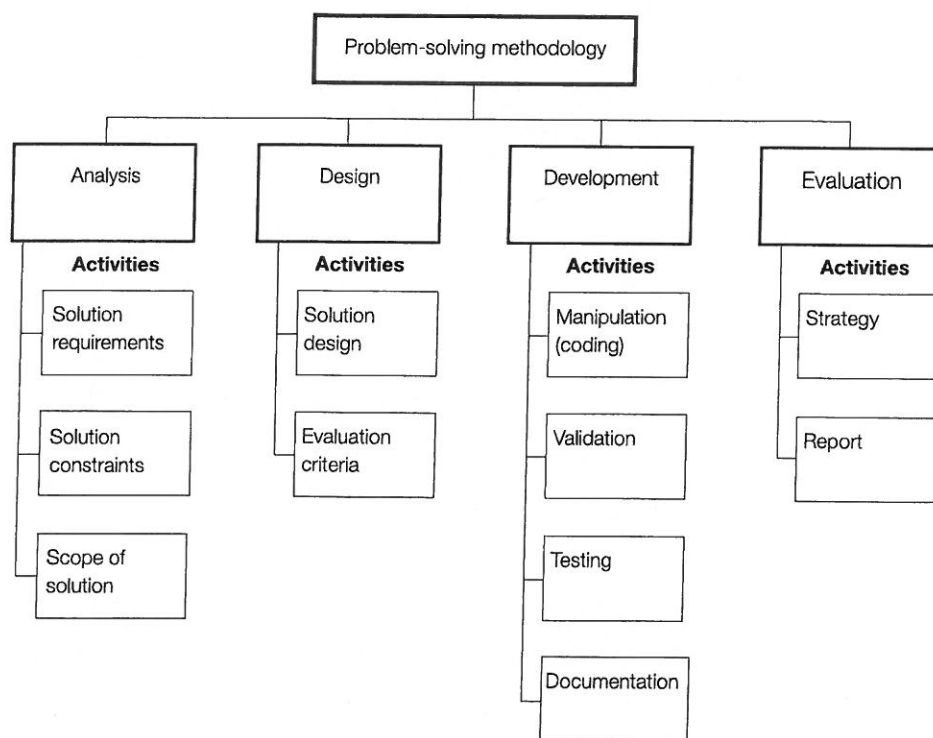


Figure 1: A problem-solving methodology for creating digital solutions

**Analysis** typically answers the ‘what questions’ – what is needed to solve a problem, given particular circumstances? It involves:

- Determining the solution requirements. What output is the solution to provide? What data is needed to produce the output? What functions must the solution provide? These requirements can be classified as being functional, that is, what the solution is required to do, and non-functional, which describes the attributes the solution should possess including useability, reliability, portability, robustness, maintainability. Tools to assist in determining the solution requirements include context diagrams, data flow diagrams and use cases.
- Identifying the constraints on the solution. What conditions need to be considered when designing a solution? Typical constraints include economic, such as cost and time; technical, such as speed of processing, capacity, availability of equipment, compatibility and security; social, such as level of expertise of users; legal, such as ownership and privacy of data requirements; and useability, such as usefulness and ease of use of solutions.
- Determining the scope of the solution. The scope states the boundaries or parameters of the solution. It identifies the area of interest or what aspects of the problem will and will not be addressed by the solution.

**Design** typically answers the ‘how questions’ – how will the solution requirements be achieved? It involves:

- Planning how the solution will function and its appearance. The solution design typically involves identifying what specific data is required and how the data will be named, structured, validated and manipulated. Typical design tools for this purpose include data dictionaries and data structure diagrams, input-process-output (IPO) charts, flowcharts, pseudocode, object descriptions. Solution design also involves, where appropriate, showing how the various components of a solution relate to one another, for example web pages, style sheets, scripts; queries, forms, reports; modules, procedures, methods, functions. Typical design tools used to show relationships include storyboards, site maps, entity-relationship diagrams, data flow diagrams, structure charts, hierarchy charts, and context diagrams.

Planning the solution also involves determining its appearance, including, where appropriate, the user interface. This typically involves identifying the position and size of text, images and graphics, font types, colours and text enhancements. Design tools used for this purpose include layout diagrams, annotated diagrams/mock ups.

- Determining the evaluation criteria. What measures will be used to judge whether or not the solution meets the requirements? These criteria should arise from the solution requirements identified in the analysis stage.

**Development** typically asks the questions of how do we realise or transform solution instructions into a working solution through the use of digital systems. It involves:

- Electronically ‘building’ or creating the solution following initial designs. It may, however, warrant modifying initial designs in order to create a working solution.
- Validation to check for the reasonableness of data being input. Validation can be both manual and electronic. Proofreading is a manual technique and it occurs when a human scans the data for errors. Electronic validation occurs when the validation process is built into the solution. Its effectiveness is determined through the testing activity.
- Testing whether the solution does what it was intended to do. This activity typically involves:
  - establishing what tests will be conducted
  - determining what test data will be used
  - determining expected results
  - conducting the test
  - recording the actual results
  - correcting any identified errors.
- Writing internal and user documentation, including within the user interface, to support the functioning and use of the solution.

**Evaluation** typically answers the question: 'How well did the solution meet its stated requirements'? It involves:

- Determining a strategy for finding out the extent to which the solution meets the required needs of the user. Typically this occurs after the solution has been developed. Usually an evaluation strategy would include specifying a timeline, outlining what data will be collected and by what methods and techniques, and how the data relates to the criteria that were generated in the design stage.
- Reporting on the extent to which the solution meets the requirements of the user. This usually takes place after the solution has been used by the user/client and is based on the criteria generated in the design stage.